

Desenvolupament d'una eina d'ajuda a correcció de pràctiques de programació

Esther Piqueras Arguisuelas

Resum

Actualment existeixen al nostre voltant milers d'eines per gestionar diferents aspectes quotidians que ens permeten estalviar temps i ser més productius. No obstant això, observem que falten aplicacions per ajudar al professorat en la realització de correccions de diferents treballs universitaris de programació, el que comporta als estudiants a una pèrdua de coneixements per manca de temps. Generalment el professorat realitza la correcció manual dels diferents treballs i dedica temps que podria emprar a pensar en una sèrie d'activitats que poden fomentar l'aprenentatge dels seus alumnes.

Donat el problema que podem observar s'ha decidit desenvolupar una aplicació d'escriptori en Java, que permeti al professorat realitzar la correcció d'una manera més automàtica, reduint així el temps emprat en aquesta. Dita aplicació pretén ser eficaç i corregir treballs (programats en C++) el més ràpid possible.

Paraules clau — Aplicació Java, aplicació d'escriptori, correcció de pràctiques.

Abstract

At the present time, there are thousand of tools in our environment to manage several everyday aspects which allow us to save time and be more productive. However, we realize that there are missing applications to assist professors in making corrections of university works programs, which means a loss of knowledge because of a lack of time. Generally, professors make manual rectifications of these several projects and spend time which could be used in preparing activities for promoting learning of their students.

Given observed problem, we decide develop Java desktop applications which allow professors make corrections automatically, reducing time spent on it. This application is intended to be effective and correct projects (programmed in C++) as quick as possible.

Index Terms — Java application, desktop application, correcting university work programs.



1 INTRODUCCIÓ

Actualment existeix un problema a l'hora de realitzar correccions per part del professorat, en diferents àmbits de programació, donat que aquestes comporten un temps molt elevat.

Generalment el professorat ha de realitzar la correcció manualment de totes les pràctiques i problemes de l'alumnat, la qual cosa vol dir que en lloc de dedicar temps a realitzar activitats d'aprenentatge als seus alumnes han de malgastar aquest temps per corregir d'una manera rudimentària les activitats que realitzen.

Donat el problema esmentat amb anterioritat, s'ha decidit realitzar aquest treball final de grau orientat al desenvolupament d'una eina d'ajuda a correccions.

2 ESTAT DE L'ART

Avui en dia hi ha diferents maneres de realitzar correccions de treballs de programació; tot i així cada eina realitza una funció diferent. Aquest projecte, com molts altres, no pretén eliminar el mètode de correcció que hi ha fins ara, només vol aconseguir automatitzar-ho per tal d'agilitzar el procés. Per poder aconseguir això s'ha de fer un estudi previ sobre com s'està realitzant actualment la correcció de les pràctiques i trobar així una millora a efectuar.

Amb l'objectiu de dur a terme el projecte segons les especificacions donades, s'ha proporcionat per part del professorat de la Universitat Autònoma de Barcelona, pràctiques de diferents alumnes per poder verificar el correcte funcionament a la fase de test, després d'implementar l'aplicació.

Aquest projecte no parteix de cap altre, sinó de la necessitat de realitzar una eina i millorar les condicions actuals que s'ofereixen als professors.

- E-mail de contacte: *esther.piqueras@e-campus.uab.cat*
- Menció realitzada: *Enginyeria del Software*
- Treball tutoritzat per: *Ernest Valveny (Departament de Ciències de la Computació)*
- Curs 2014/15

3 OBJECTIUS

El principal objectiu és desenvolupar una eina d'ajuda a correcció de pràctiques de programació. Concretament, realitzar una aplicació d'escriptori pel professorat amb la qual es pugui ajudar a la correcció de les diferents activitats d'una manera més eficient.

Aquesta aplicació estarà implementada en Java i es podrà utilitzar per realitzar correcció d'activitats de programació realitzades en el llenguatge C++.

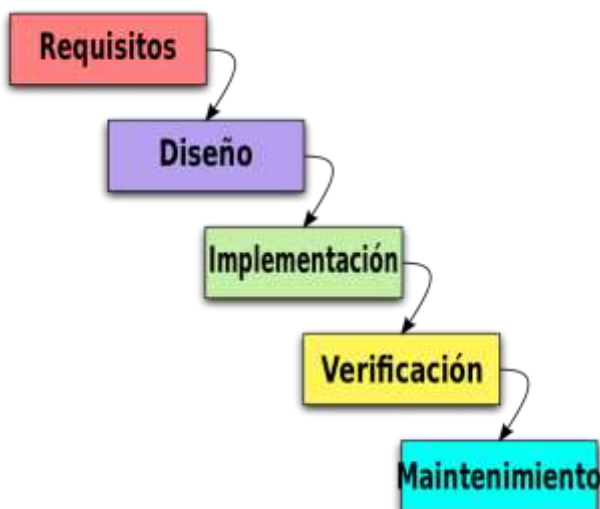
Aquest objectiu és el més important, no obstant això, no podem oblidar els objectius que ha de satisfer l'aplicació. Aquests seran esmentats a continuació:

- Ha de permetre avaluar el programa des de dos punts de vista: funcionament i correcció del codi.
- Ha de permetre executar un programa i validar la seva sortida.
- Ha de permetre visualitzar fàcilment parts predefinides del codi.
- S'han de generar automàticament les qualificacions finals.

4 METODOLOGIA

Existeixen diversos mètodes per portar a terme aquest projecte, encara que la majoria són mètodes per equips de desenvolupament amb un nombre elevat d'integrants.

En aquest cas, al ser un projecte final de grau els únics que intervenen són l'alumne i el professor amb uns objectius clarament definits des d'un inici. Per tot això, s'ha utilitzat el mètode en cascada, que ens proporciona una divisió de fases i seguiment, adequat per aquest projecte.



Imatge 1: Metodologia en Cascada.

La metodologia en cascada consta de 5 fases:

- 1.- Requisits
- 2.- Disseny
- 3.- Implementació
- 4.- Verificació
- 5.- Manteniment

Aquest cas constarà de 4 fases donat que no hi haurà fase de manteniment de l'aplicació, encara que si es podran realitzar millores. Per tant, la metodologia a seguir durant aquest projecte està dividida en les següents fases:

- 1.- Presa de Requisits
- 2.- Disseny
- 3.- Implementació
- 4.- Verificació

Gràcies a dur a terme una metodologia com aquesta, s'ha pogut acabar el projecte en un temps fixat correcte i, encara que han pogut sorgir petits inconvenients, s'han pogut solucionar en la mateixa fase sense realitzar un canvi general i redistribució del temps.

Com qualsevol metodologia, aquesta té avantatges i inconvenients que faciliten el procés de desenvolupament del projecte. Han estat més els avantatges que els inconvenients i, gràcies a aquestes, tot ha sortit com s'esperava en un principi.

4.1 Avantatges

- És molt adequada per equips petits que necessiten treure ràpidament el producte al mercat, ja que és el que s'espera a l'hora de realitzar un treball de final de grau.
- És un model fàcil d'implementar i entendre, el qual permet realitzar tot de manera més eficient.
- És un model conegut i utilitzat amb freqüència, el que proporciona un millor feedback en les reunions amb el tutor.
- Promou una metodologia de treball efectiu, donat que totes les fases són correlatives, i això fa que sorti tot com s'espera.

4.2 Inconvenients

- Normalment els projectes no són lineals i això fa que si volem utilitzar aquesta metodologia, anirà de la mà amb un altre.
- Qualsevol errada en les fases finals condueix a una revisió des de la fase de disseny.
- Una fase no es pot portar a la fi sense acabar la fase anterior i definida la següent.

4.3 Metodologia VS Projecte

El fet d'utilitzar aquesta metodologia, com ja he esmentat anteriorment, ha fet que pogués realitzar el projecte correctament. Aquesta metodologia a pesar de tenir inconvenients, han aportat més avantatges que altra cosa.

Un dels inconvenients és el fet d'acabar una fase abans de començar un altre, però en aquest cas, he fet fases una mica en paral·lel com s'explica en la part de planificació, i això en aquest cas ha estat un avantatge.

Sí que és veritat que un dels inconvenients que més m'han afectat en utilitzar la metodologia ha estat en la fase d'implementació. A mesura que anava realitzant la implementació, m'adonava que sortien petits inconvenients que eren una errada a l'hora de pensar com es dissenyarien. A pesar d'aquests inconvenients i al ser un projecte lineal, no ha afectat greument a la realització d'aquest.

5 PLANIFICACIÓ

Per tal de portar a terme el treball en el temps previst, s'ha realitzat una petita planificació del temps que hi ha en relació al treball que s'ha de realitzar.

Nombre	Duració	Inici	Fin
Treball de Final de Grau	98d	12/02/2015	29/06/2015
Preses de Requisits	11d	12/02/2015	26/02/2015
Disseny	16d	26/02/2015	19/03/2015
Implementació	41d	19/03/2015	14/05/2015
Test	26d	14/05/2015	18/06/2015
Realització de l'article	6d	18/06/2015	25/06/2015
Preparació de l'exposició	2d	25/06/2015	26/06/2015
Finalització del treball de Final de Grau	1d	29/06/2015	29/06/2015

Imatge 2: Planificació

A continuació es mostra en els següents apartats una petita explicació de cada fase principal enllaçada amb la metodologia emprada:

5.1 Fase 1. Presa de requisits

Aquesta fase té una duració d'11 dies en els quals hi ha una presa de requisits amb el tutor i queden ben definits i clars els objectius concrets de l'aplicació. Aquests objectius en un principi són invariants, encara que sí que es poden realitzar millores si s'escau convenient.

Els requisits s'han anat recollint en les diverses reunions que s'han realitzat entre el tutor i l'alumne, per tal d'aclarir dubtes que poguessin sorgir abans de realitzar la implementació del projecte.

Tots aquests requisits recopilats durant les primeres reunions s'han formalitzat en una reunió final en la qual es

van donar per bons tots, i és aleshores, va ser quan es va començar la següent fase.

5.2 Fase 2. Disseny

La fase de disseny comporta els darrers 16 dies a la fase de requisits i serveix per posar en comú amb el tutor una interfície gràfica visualment fàcil per l'usuari.

En aquesta fase és primordial realitzar un disseny que compleixi amb tots els requisits pactats prèviament. Per tant, podem dir que es fa el disseny bàsic necessari per poder començar la implementació i, aquest, sempre podrà tenir canvis amb posterioritat de la seva definició, encara que no és l'esperat.

Aquesta fase dura molt poc temps donat que no és una fase primordial, ja que cap objectiu definit era fer un bon disseny de l'aplicació. Per aquest motiu la següent fase d'implementació serà més llarga, ja que és on es realitza la implementació de tots els objectius fixats.

5.3 Fase 3. Implementació

La fase que comporta més temps d'un projecte és la fase d'implementació. En aquest cas ocupa 41 dies en els quals es desenvolupa l'aplicació amb el compliment dels requisits i el disseny pactat amb anterioritat.

És una fase en la qual s'ha de recopilar tot el treballat fins ara i posar en pràctica tot l'aprenent a la universitat. Es podria dir, aleshores, que és la fase crítica de tot el projecte, juntament amb la fase de verificació, on es realitzarà el test de l'aplicació feta.

A mesura que es produeix la fase d'implementació, es va introduint la de verificació, ja que cada petita implementació que es desenvolupa s'ha de verificar per donar-la com bona. Això és a causa de la metodologia que estem emprant, la metodologia en cascada. Si fos un altre tipus de metodologia com l'espiral, aleshores aquestes dues fases no anirien conjuntament per realitzar aquest projecte.

5.4 Fase 4. Verificació

L'última fase, però no menys important, que ocupa al voltant de 26 dies, és la de verificació.

En aquesta fase es realitzaran diferents proves per tal de comprovar que l'aplicació funcioni correctament i compleixi tots els objectius predefinits. Per això diem que es realitza conjuntament amb la fase anterior d'implementació.

En aquest moment és quan s'utilitzen els diferents recursos proporcionats pel tutor com són els problemes i projectes de programació d'altres alumnes.

Gràcies a aquests, aquesta fase s'ha pogut realitzar sense massa problemes i s'han anat modificant petites coses implementades que donaven lloc a errors a l'hora de complir els requisits.

6 FUNCIONAMENT

Per tal d'automatitzar el més possible la correcció de pràctiques, es realitza l'aplicació explicada a continuació. L'aplicació consta de 5 parts clarament definides, que coincideixen amb les pantalles de l'aplicació:

- 1.- Configuració.
- 2.- Test Manual.
- 3.- Test Automàtic.
- 4.- Visualització de codi.
- 5.- Desar notes.

El funcionament de l'aplicació és molt senzill. Comença amb la part de configuració, i un cop tenim els fitxers carregats es continua amb un dels 3 mètodes de correcció (Test Manual, Test Automàtic o Visualització del Codi) i, per últim, es guarden les notes. En els darrers punts d'aquest apartat es posaran de manifest les diferents accions que comporta cada part.

6.1 Configuració

La part de configuració tracta de la càrrega d'un fitxer redactat per l'usuari, que no ha de tenir cap nom en concret. En aquest fitxer es trobarà primer de tot la ubicació de la carpeta que conté les activitats, problemes o projectes a corregir. A continuació, el llistat de les funcions que es volen corregir, introduït manualment per l'usuari, i al costat de cada funció, separat per un tabulador el percentatge amb el qual es vol valorar la correcció d'aquella funció dins la pràctica completa.

Per comprovar que la càrrega del fitxer ha estat correcte i que s'ha introduït el fitxer bo, l'aplicació mostra per pantalla el contingut del fitxer. D'aquesta manera, l'usuari pot rectificar el fitxer a utilitzar durant l'execució de l'aplicació.

Un cop està feta la configuració, aquesta càrrega el primer fitxer a corregir; per tant, l'usuari pot accedir al test manual, automàtic o a visualitzar el codi per començar la correcció d'aquella activitat.

Si es vol afegir alguna llibreria extra per poder compilar algun projecte o pràctica en concret, es necessari indicar el path on es troben les llibreries. Aquest path s'introdueix manualment a l'apartat de llibreries que trobem en la pantalla de configuració en la part de sota.

6.2 Test Manual

Quan es troba carregat el fitxer a avaluar, amb les funcions predefinides per corregir, l'usuari pot escollir l'opció de test manual.

Aquesta opció presenta una finestra emergent en la qual es realitza l'execució del treball a avaluar, i d'aquesta manera, l'usuari pot anar puntuant sota el seu criteri.

Un cop finalitzada la seva interacció amb el programa, es poden desar les dades de qualificacions i aquestes, posteriorment, es mostraran en la pantalla de notes.

6.3 Test Automàtic

La realització del test automàtic és molt més senzilla que la manual. En la carpeta on es troben els fitxers a corregir es pot introduir un fitxer amb un llistat d'execucions que es volen fer. Aquest fitxer ha de tenir el nom `execucion.txt` i variarà segons el projecte o activitat a corregir.

Si se selecciona el test automàtic, aquest fitxer es carrega i executa les comandes que hi ha dins de manera que la seva sortida es mostra per pantalla a l'apartat resultats dins de la pantalla anàlisi. Aquesta opció de l'aplicació es duu a terme amb problemes més que amb projectes de programació dona que els problemes normalment son petites entregues que no necessiten interacció.

Un cop s'ha realitzat l'execució, com en el test manual, es carreguen les qualificacions en l'apartat de notes.

6.4 Anàlisi visual de l'activitat

L'anàlisi permetrà visualitzar les funcions que es poden corregir de l'activitat, problema o projecte carregat.

Per tal de realitzar això d'una manera el més fàcil possible, s'ofereix a l'usuari un llistat de totes les funcions disponibles a visualitzar per poder escollir.

Aquestes funcions són les que es troben al fitxer de configuració prèviament carregat i escrit per l'usuari, donat que segons el que es vol corregir, les funcions seran diferents.

Un cop es té seleccionada la funció que es vol corregir, es mostrarà a la pantalla el codi corresponent, per poder així, ser puntuada pel professor.

Com a l'opció del test manual i automàtic, la qualificació que es guardi serà visualitzada posteriorment a la pantalla de notes.

6.5 Desar Notes

Aquesta opció és l'última abans de guardar les qualificacions finals en un fitxer de sortida, el qual no es podrà modificar més dins de l'aplicació. Les notes si es podran modificar si s'accedeix al arxiu excel posteriorment, però mai es podran modificar un cop guardem en aquesta pantalla, donat que aquesta acció carrega els fitxers d' uns altres alumnes.

Es tracta d'una pantalla que conté les notes que s'han anat ponderant al llarg de la correcció del treball des de l'aplicació. En aquest moment, l'usuari comprova que totes les notes siguin correctes per aquell treball.

Si és així, desarà les notes en un .csv on es mostrarà el nom de l'alumne, el niu i les seves qualificacions. Si no es creu correcte, es pot modificar abans de desar.

Un cop desades les notes automàticament, l'aplicació carrega el nou fitxer a corregir.

7 RESULTATS

Després de 98 dies dedicats a la realització del projecte, podem dir que hi ha hagut un resultat bo i semblant a la idea proposada en un principi. El més important de tot és que compleix tots els requisits i objectius que s'han demanat en un principi.

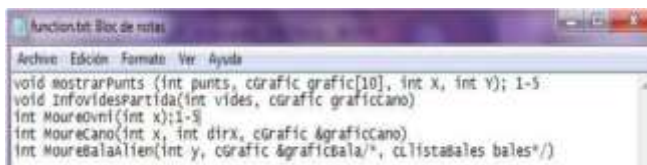
El projecte comença amb una pantalla d'introducció amb un disseny molt bàsic en el qual podem accedir a l'aplicació o al manual d'explicació d'aquesta.



Imatge 3: Pantalla Inicial

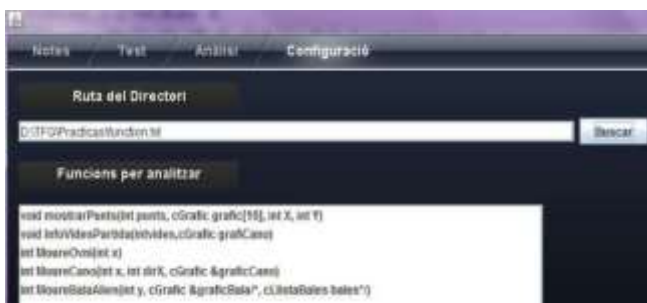
En aquesta pantalla, com es pot observar, s'han utilitzat els colors característics de la Universitat Autònoma de Barcelona. És molt senzilla i permet a l'usuari una interacció ràpida, ja que s'utilitzen els símbols als quals estem acostumats a veure per realitzar les diferents accions d'ajuda o run.

A continuació, haurem de realitzar el fitxer de configuració que es carregarà a l'aplicació. Aquest document que no ha de tenir cap nom específic és un simple .txt amb la ubicació dels fitxers a tractar.



Imatge 4: Exemple de fitxer de configuració.

Com ja s'ha esmentat anteriorment, l'usuari en carregar el fitxer ha de poder observar si és correcte, per tant l'aplicació mostrarà a la pantalla de configuració el fitxer carregat. Un exemple de com funciona és el següent:

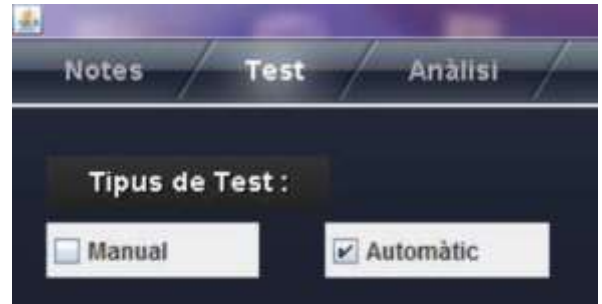


Imatge 5: Pantalla de configuració.

Un cop tinguem la configuració realitzada, automàticament es carrega el primer fitxer que es pot corregir.

Aquest fitxer, com ja hem dit, té diverses opcions de correcció (manual o automàtica) i, d'altra banda, la visualització del codi.

Seguidament, es posen un seguit d'imatges per veure el funcionament de l'aplicació amb relació a la correcció de les pràctiques.



Imatge 6: Pantalla de test.

Aquesta pantalla ens donaria l'opció de carregar el fitxer de correcció automàtic i, per tant, executar el resultat, o bé, executar el treball de programació en una finestra emergent en escollir l'opció manual.

Per últim, com a mètode de correcció, es mostra la pantalla de visualització del codi en dues captures de pantalla. La primera mostra les opcions a escollir que ens donen de la funció que volem corregir i la segona, el codi.



Imatge 7: Pantalla d'anàlisi. Funcions a escollir



Imatge 8: Pantalla d'anàlisi. Codi

En aquest moment ja s'ha realitzat la correcció del treball de programació i es processen les qualificacions generades i guardades.

	A	B	C	D	E	F	G	H
1	Nom	NÚ	vold mostrarPru vold InfoVid int MaureDev(int x) int MaureCa int MaureRe Total					
2	Guillem Villanoya Morales	138032	3	5	3	3	3	3,8
3	Guillem	138044	3	5	3	3	3	3,8
4	Germán González Fuentes	139130	6	6	3	5	6	5
5	Oscar	139583	6	6	2	5	6	5
6	David Lopez Zapate	139197	7	6	8	8	5	6,8
7	Arnau Casale	138094	7	6	8	8	5	6,8

Imatge 9: Resultat de l'execució completa.

Ara ja s'han corregit totalment els fitxers situats en la carpeta a processar i podem visualitzar totes les notes generades per tots els treballs entregats.

En obtenir com a resultat un .csv, aquest igualment es podrà modificar i entregar als alumnes per poder comprovar la seva puntuació respecte al treball entregat.

8 POSSIBLES MILLORES

Durant el desenvolupament del projecte m'he adonat que es podrien realitzar diverses millores sobre aquest, i segurament si ho comentéssim amb diferents professors de programació, segurament hi hauria moltes millores més a realitzar.

Encara que cal dir que el projecte realitza els objectius que es van definir a l'hora d'implementar-lo.

M'hauria agradat realitzar les següents millores, però per temps ha estat impossible. No obstant això, són millores que es podrien realitzar en altres treballs de final de grau, i per tant les comparteixo a continuació:

1.- Introduir el llistat de funcions de manera automàtica.

Actualment el llistat de funcions es carrega a partir del fitxer de configuració donat per l'usuari. Aquesta càrrega del fitxer es podria eliminar realitzant la càrrega automàtica de les funcions.

La idea és donar directament la ubicació dels fitxers a corregir (no llegir-la del fitxer com ara) i carregar el primer per corregir. En aquest moment es podria fer una lectura del fitxer i detectar les funcions que es poden trobar, per guardar-les en un arxiu o una variable simplement.

D'aquesta manera, es podria carregar la variable o l'arxiu per mostrar el llistat de funcions i no haurien d'introduir manualment les funcions l'usuari.

Aquesta és una millora molt simple però que s'ha pensat després de realitzar el projecte i no quedava més temps per implementar-la.

2.- Web per mostrar les notes a l'alumne de manera automàtica.

Es tractaria de realitzar una pàgina web interna per l'alumne, d'aquesta manera el professor no intervindria si no fos necessari.

La idea principal és que mitjançant aquesta web l'alumne carregui la seva pràctica i obtingui la nota resultant. Per fer això, només s'hauria de realitzar la pàgina web tenint en compte els permisos de qui pot accedir, entre altres coses.

Posteriorment, l'alumne carregaria el fitxer de la pràctica i aquest seria enviat a l'aplicació de correcció implementada en aquest treball. Un cop l'aplicació realitzés la seva execució sobre la pràctica, retornaria la nota, que hauria de ser visualitzada per l'alumne a través de la plataforma.

Cal dir que per implementar aquesta millora s'hauria de modificar pot ser alguna cosa de la implementació de l'aplicació resultant avui dia, però seria interessant per automatitzar més, ja que proporcionaria dos avantatges molt clars:

- Menys treball per part del professorat, el que resultaria més temps per poder preparar activitats per l'aprenentatge de l'alumnat.
- Més rapidesa a l'hora de poder visualitzar la nota resultant del treball per l'alumnat, ja que actualment com es fa manual, l'alumnat triga molt a obtenir les seves qualificacions d'un simple treball.

3.- Realitzar l'aplicació per correcció de diferents llenguatges de programació

Després de realitzar aquest projecte s'obté una aplicació que realitza la correcció de treballs de programació fets amb el llenguatge de C++.

Per realitzar aquesta millora, només s'hauria de realitzar un petit canvi a l'hora de cridar al compilador i fer la correcció en la part de test manual i automàtic.

La resta de l'aplicació no varia, ja que per visualitzar el codi no importa el llenguatge amb el qual estigui realitzat i per desar les notes tampoc.

Aquest és un petit canvi fàcil de realitzar i que implicaria que es poguessin corregir més pràctiques i facilitar, per tant, la correcció de treballs a diferents professors.

4.- Realitzar un projecte per correcció d'exàmens de programació.

Aquesta idea realment no és una millora sobre el projecte realitzat, sinó una idea sorgida a partir d'aquest projecte.

Si actualment es corregeixen pràctiques de programació, també es podria realitzar correcció d'exàmens, i no només de programació, sinó qualsevol.

Gràcies al reconeixement de dades, es podria realitzar un projecte de final de grau bastant interessant i, sobretot, un projecte que realment automatitzaria el procés i faria molt més fàcil la correcció de qualsevol treball al professorat.

9 CONCLUSIÓ

Els objectius originals s'han complert correctament, implementant les funcionalitats definides en un principi. Per poder aconseguir arribar a complir els objectius i finalitzar correctament el projecte, s'han anat fent un seguit de processos de l'enginyeria del software que han permès desenvolupar el projecte de la millor manera possible.

Durant la primera fase s'han anat treballant els diferents conceptes adquirits al llarg de la carrera per poder realitzar una bona presa de requisits. De cara al projecte, va ser molt important el fet de realitzar una presa real de requisits i fixar els objectius clars de l'aplicació, ja que amb la metodologia emprada i el temps que hi ha no es podia permetre cap error.

La fase de disseny va ser una mica més variant i no tant estricta, ja que el disseny no era el més important per l'aplicació. Aquesta fase va ser una mica conjunta amb la fase d'implementació perquè necessitava part del disseny per poder implementar, però aquest podia modificar-se més tard sense problemes.

D'altra banda la fase d'implementació va ser més costosa, no tant per les tecnologies que s'han utilitzat sinó pel temps proporcionat. Durant aquesta fase, he pogut posar en pràctica els coneixements adquirits fins ara adonant-me que realment durant la carrera s'assoleixen coneixements que es poden posar en pràctica a la vida real.

Per últim, la fase més fàcil ha estat la de test on he pogut comprovar com a professor les errades que cometen els alumnes a l'hora de realitzar entregues.

Pel que fa a les tecnologies utilitzades, tant el llenguatge emprat com l'entorn de desenvolupament, eren coneguts i això va facilitar molt el treball. Per altra banda era complicat pel fet de no ser un projecte proposat per mi.

Tot el projecte s'ha anat desenvolupant pràcticament sol, sense imprevistos que no es poguessin controlar, ni problemes impossibles de solucionar. El conjunt de tot ha estat el que m'ha permès finalitzar el treball sense complicacions i en el temps previst des d'un principi.

AGRAÏMENTS

M'agradaria agrair al meu tutor acadèmic, Ernest Valveny, pel seu ajut durant tot el desenvolupament del projecte i la seva comprensió durant l'última fase d'aquest, ja que no he pogut estar 100% per temes de treball.

D'altra banda, també m'agradaria donar les gràcies a Aureli Bisbe i Rubén Guix, dos bons companys que m'han recolzat al llarg de tota la carrera. Sense ells no hagués viscut anys universitaris tant divertits, encara que durs. Gràcies a ells he pogut saber la definició de TOP-DOWN, el significat de companyerisme, de dedicació,... i sobretot el fet de seguir endavant sigui quina sigui la situació.

També a Arcadi Llanza o Cristina Diéguez pel seu ajut i distracció en els moments difícils de portar, al llarg d'aquest treball. Així com els esmorzars diaris a la universitat que serveixen per la creació de nos idees que s'implementaran al sortir de la carrera.

I per últim, donar les gràcies a persones com David Garcia, i la seva família, que és com la meua, que han fet que tot fos més fàcil i suportable i mai han deixat que caigués. I sobretot, al meu pare, que en temps durs sempre ha estat al meu costat, plorant, rient, i demostrant que sempre es pot sortir endavant amb tot.

BIBLIOGRAFIA

- [1] Vídeos de com realitzar una aplicació [online] visitat en 2015, disponible en:
<http://javax-peru.blogspot.com.es/2010/04/video-tutorial-desarrollo-de.html>
- [2] Llenguatge Java [online] visitat en 2015, disponible en:
<http://www.chuidiang.com/java/index.php>